

CLAIMS

What is claimed is:

1. A computer-implemented method for monitoring a computer system when said computer system executes a user application using a production operating system (OS), said OS having a OS kernel with a kernel trap arrangement, said method comprising:

providing a diagnostic monitor, said diagnostic monitor being configured to be capable of executing even if said OS kernel fails to execute, said diagnostic monitor having a monitor trap arrangement; and

if a trap is encountered during execution of said user application, ascertaining using said diagnostic monitor whether said trap is to be handled by said OS kernel or said diagnostic monitor; and

if said trap is to be handled by said OS kernel, passing said trap to said OS kernel for handling.

2. The computer-implemented method of claim 1 wherein said OS kernel is a production kernel configured to execute said user application.

3. The computer-implemented method of claim 2 wherein said OS kernel is configured to permit said diagnostic monitor to be initialized prior to loading said OS kernel.

4. The computer-implemented method of claim 1 further comprising:
ascertaining, using said diagnostic monitor, if said trap is to be handled by said diagnostic monitor, whether said trap represents a monitor call by said OS kernel; and
if said trap represents said monitor call by said OS kernel, using said diagnostic monitor for handling said monitor call.

5. The computer-implemented method of claim 4 further comprising using said diagnostic monitor for performing, if said trap represents said monitor call by said OS

kernel, a) sending a freeze signal to any other processor in said computer system, b) collecting state information, and c) performing analysis on said state information.

6. The computer-implemented method of claim 5 further comprising using said diagnostic monitor for performing, if said trap represents said monitor call by said OS kernel, d) displaying a result of said analysis in a shell, and e) receiving user input, if any, from said shell.

7. The computer-implemented method of claim 4 further comprising:
if said trap does not represent said monitor call, using said diagnostic monitor for performing a) sending a freeze signal to any other processor in said computer system, b) collecting state information, and c) performing analysis on said state information.

8. The computer-implemented method of claim 7 further comprising using said diagnostic monitor for performing, if said trap does not represent said monitor call, d) displaying a result of said analysis in a shell, and e) receiving user input, if any, from said shell.

9. The computer-implemented method of claim 1 further comprising:
converting, using said OS kernel, a panic call into a monitor call;
ascertaining, using said diagnostic monitor, whether said monitor call represents said panic call when said monitor call is received by said diagnostic monitor;
if said monitor call represents said panic call, using said diagnostic monitor for performing a) sending a freeze signal to any other processor in said computer system, b) collecting state information, and c) performing analysis on said state information.

10. The computer-implemented method of claim 9 further comprising using said diagnostic monitor for performing, if said monitor call represents said panic call, d) displaying a result of said analysis in a shell, and e) receiving user input, if any, from said shell.

11. The computer-implemented method of claim 1 wherein said OS kernel is configured to generate a trap responsive to a time-out event of a kernel timed semaphore.
12. The computer-implemented method of claim 1 wherein said diagnostic monitor is configured to continue executing after said OS kernel crashes.
13. The computer-implemented method of claim 1 wherein said diagnostic monitor is configured to continue executing even if said OS kernel fails to load.
14. The computer-implemented method of claim 1 wherein said diagnostic monitor is configured to narrow down an error encountered during said execution of said user application to a field-replaceable unit (FRU).
15. In a computer system, an arrangement for diagnosing a computer system while executing a user application program, said user application program being executed under a production operating system (OS) kernel, comprising:

a diagnostic monitor configured to execute cooperatively with said OS kernel, said OS kernel having a kernel trap arrangement for handling at least one of a trap-type message and an interrupt-type message generated during said execution of said application program, said diagnostic monitor being capable of continuing to execute even if said OS kernel fails to execute, said diagnostic monitor including a monitor trap arrangement for handling at least one of said trap-type message and said interrupt-type message, said diagnostic monitor being configured to receive traps generated during said execution of said application program and decide, for a given trap received, whether said OS kernel would handle said trap received or whether said diagnostic monitor would handle said trap received.
16. The arrangement of claim 15 further comprising a loader configured to load said diagnostic monitor prior to loading said OS kernel at system initialization.
17. The arrangement of claim 15 wherein said diagnostic monitor includes monitor logic and a monitor library.

18. The arrangement of claim 17 wherein said monitor library includes computer-implemented code for resolving error conditions to a set of field replaceable units.

19. An article of manufacture comprising a program storage medium having computer readable code embodied therein, said computer readable code being configured to handle errors in a computer system, comprising:

computer readable code implementing a diagnostic monitor, said diagnostic monitor being configured to execute cooperatively with a production operating system (OS) kernel in said computer system and configured to be able to execute even if said OS kernel fails to execute, said diagnostic monitor including a monitor trap arrangement configured to receive traps generated in said computer system, said OS kernel having a kernel trap arrangement configured to handle traps passed from said diagnostic monitor; and

computer readable code for loading said diagnostic monitor at system startup prior to loading said OS kernel.

20. The article of manufacture of claim 19 wherein said diagnostic monitor is configured to isolate an error generated during execution of an application program executing under said OS kernel to a set of field replaceable units in said computer system.

21. The article of manufacture of claim 19 wherein said computer readable code implementing said diagnostic monitor includes computer readable code implementing monitor logic and computer readable code implementing a monitor library.

22. The article of manufacture of claim 21 wherein said computer readable code implementing said monitor library includes computer implemented code for resolving error conditions to a set of field replaceable units.